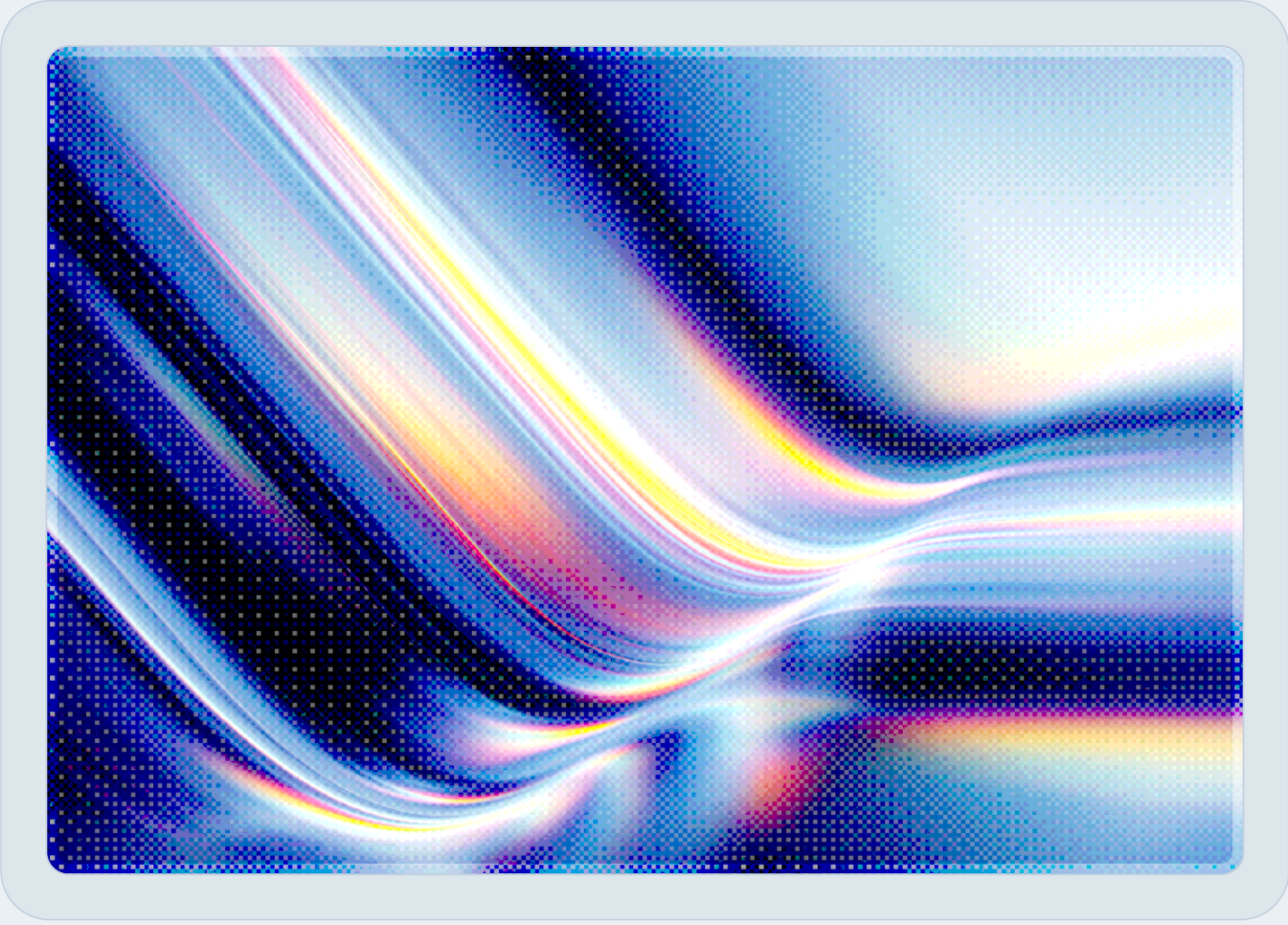
CSAI Foundation | Cloud Security Alliance

36,769 Exposed AI Endpoints: A Self-Hosted Supply Chain Crisis

Unauthenticated Model Servers, Agent Platforms, and Vector Databases on the Open Internet

2026-09-15

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- Internet-scanning research published in September 2026 identified 36,769 self-hosted AI endpoints — model servers, agent-building platforms, and vector databases — that were reachable from the public internet and identifiable by application fingerprint, with the researchers explicitly describing the figure as a lower bound rather than a full census [1].
- Only 2.02% of the endpoints returned an HTTP authentication challenge, meaning the overwhelming majority had no access gate whatsoever standing between an anonymous internet user and a running AI system [1].
- Open WebUI, which had the largest number of internet-exposed instances in the dataset, accounted for 18,529 of the reachable instances; only one required authentication [1].
- Research by GitGuardian, reported in the same Security Affairs coverage, on exposed workflow and agent-orchestration tools found 4,576 unique n8n API tokens leaked on GitHub, of which 321 of 896 tested reachable instances accepted the leaked credentials outright [1].
- The exposure pattern documented here is consistent with, and extends, prior CSA AI Safety Initiative findings on unauthenticated Model Context Protocol (MCP) servers and AI gateway infrastructure, reinforcing that insecure defaults — not sophisticated exploitation — remain a primary driver of risk in self-hosted AI deployment [2][3].

Background

Self-hosted AI tooling has grown popular over roughly the past two years: local inference servers such as Ollama, vLLM, LocalAI, and llama.cpp; browser-based front ends such as Open WebUI; and low-code agent and workflow builders such as Flowise, n8n, Dify, Langflow, and ComfyUI. These tools lower the barrier to running large language models and building AI-powered automations without depending on a hosted SaaS provider, and organizations and individual developers alike have adopted them accordingly. The scanning research described below suggests that, for a meaningful share of these deployments, authentication and network-exposure controls have not kept pace. Many of these projects ship with authentication disabled or optional by default, on the assumption that deployments will sit behind a private network, a reverse proxy, or a VPN — an assumption that internet-scanning research has repeatedly shown does not hold in practice.

In September 2026, researchers at Mysterium VPN published a study that queried Netlas, a third-party internet-scanning index, for application fingerprints associated with self-hosted AI software [1]. Rather than connecting to, exploiting, or retrieving data from any exposed system, the researchers counted fingerprint matches and used the presence or absence of an HTTP 401 or 403 authentication challenge as a proxy for whether an endpoint was gated. This methodology deliberately avoided touching live systems or publishing identifying details such as IP addresses, but it produced a total of 36,769 distinct AI-related services reachable from the open internet, the large majority of them answering requests with no authentication barrier at all.

This finding does not exist in isolation. It follows a string of CSA AI Safety Initiative advisories documenting similar exposure patterns in adjacent AI infrastructure — publicly accessible, unauthenticated MCP server instances [3], and a chained, KEV-listed vulnerability in the LiteLLM AI gateway that was reachable specifically because operators had exposed it to the internet without a protecting proxy [2]. The Mysterium VPN findings extend this picture downstream, from AI gateways and protocol servers to the inference engines, agent builders, and vector stores that sit at the foundation of the self-hosted AI stack. Read together, these findings describe a consistent failure mode across the AI supply chain: software that treats authentication as an optional, deployment-time decision rather than a secure-by-default requirement, deployed in configurations where reachability from the public internet may not have been an intentional or reviewed decision.

Security Analysis

The Mysterium VPN dataset breaks the exposed population into three broad categories, summarized below.

Category	Representative Software	Reachable Instances	Authenticated
Model inference servers	Open WebUI	18,529	1
Model inference servers	Ollama	6,935	Not reported
Model inference servers	vLLM	4,880	3

Category	Representative Software	Reachable Instances	Authenticated
Model inference servers	LocalAI	150	0
Model inference servers	llama.cpp	69	0
Agent/workflow platforms	Flowise, n8n, ComfyUI, Dify, RAGFlow, Langflow	5,223	Not reported in aggregate; Flowise subset (1,341 instances) reported 0 authenticated
Vector databases	Milvus/Attu consoles	920	Not reported

Source: [1]. Authentication figures reflect HTTP 401/403 challenges observed by the researchers; absence of a challenge does not confirm exploitability, only absence of a network-layer gate. These row figures are drawn from the individual per-product and per-category counts reported in the source article; they sum to 36,706, sixty-three short of the 36,769 total the researchers report for the full scan. The source presents these figures narratively rather than as a formal, summable table, so the residual instances likely reflect additional fingerprinted products or categories not broken out individually here, rather than an error in any single row.

Each category carries a distinct risk profile. Exposed inference servers such as Ollama and vLLM allow anonymous users to consume GPU and compute resources at the operator's expense — a pattern some security researchers have termed "LLMjacking" — and, depending on configuration, may expose loaded model artifacts, request logs, or system prompts to any visitor. Open WebUI's dominance in the dataset is notable precisely because it is designed as a user-facing chat interface: an unauthenticated instance can expose conversation history, uploaded documents, and connected knowledge bases to anyone who finds it, and the project's own documentation explicitly warns operators against exposing it directly to the public internet without an additional access-control layer [4].

CSA assesses that agent and workflow platforms present a comparatively sharper risk, since they are commonly used to store credentials for downstream integrations — email accounts, cloud APIs, databases, and other AI services — inside workflow definitions. Research by GitGuardian, reported in the same Security Affairs coverage, found 4,576 unique n8n API tokens exposed through GitHub search, and when those leaked tokens were tested against the 896 reachable n8n instances fingerprinted in the Mysterium VPN scan, 321 accepted the credentials [1]. This is a distinct and arguably more severe failure mode than a missing authentication gate: it demonstrates that credential hygiene failures (secrets

committed to public repositories) and infrastructure exposure failures (internet-reachable management consoles) compound each other, turning a leaked token into a live foothold inside an organization's automation pipeline. Compounding this, Flowise — one of the agent-builder platforms represented in the dataset, none of whose 1,341 reachable instances returned an authentication challenge — was separately disclosed in 2026 to contain CVE-2026-40933, a critical (CVSS 9.9) command-injection vulnerability in its MCP adapter that allows an authenticated attacker to achieve arbitrary remote code execution on the underlying host by abusing an allowlisted command in the platform's custom MCP configuration interface [5]. An internet-exposed Flowise instance with a weak or default credential would clear the CVE's "authenticated attacker" precondition trivially; while this research did not document leaked Flowise credentials specifically, the n8n token-reuse findings above illustrate how commonly credential hygiene failures compound infrastructure exposure in this class of tooling.

Vector databases rounded out the dataset at a smaller but still consequential 920 exposed instances, predominantly Milvus deployments reachable through their Attu administrative console. Vector stores back retrieval-augmented generation (RAG) pipelines and frequently contain embeddings derived from internal documents, customer data, or proprietary knowledge bases; an exposed administrative console can allow an anonymous user to query, export, or manipulate that underlying content without ever touching the inference layer itself.

The scale problem this research documents is not unique to the Mysterium VPN methodology. Bishop Fox's AIMap, an open-source AI attack-surface discovery tool released in May 2026, was built to address the same underlying trend: publicly reachable Ollama servers, MCP endpoints, and inference proxies that have "multiplied across the internet over the past year, often deployed without authentication or rate limits" [6]. AIMap fingerprints a wider set of protocols — including MCP, LiteLLM, LangServe, OpenClaw and Clawdbot systems, Gradio, Streamlit, and HuggingFace TGI — and scores each discovered endpoint on authentication posture, tool exposure, and other risk factors, reinforcing that the Mysterium VPN count of 36,769 should be read as a lower bound within a larger self-hosted AI attack surface than this single scan captures, rather than a comprehensive figure [1][6].

Recommendations

Immediate Actions

Security teams should treat any self-hosted AI inference server, agent-builder platform, or vector database as they would any other internet-facing production system, starting with an inventory question: is this service reachable from the public internet, and if so, was that reachability intentional? Operators can check their own deployments against the application fingerprints published in the underlying

research or run an open-source discovery tool such as AIMap against their own address ranges to identify unintentionally exposed instances [1][6]. Any Ollama, vLLM, LocalAI, llama.cpp, Open WebUI, Flowise, n8n, ComfyUI, Dify, RAGFlow, Langflow, or Milvus/Attu deployment found reachable without authentication should be bound to localhost or a private network interface immediately, or placed behind an authenticated reverse proxy if external access is genuinely required. Organizations running Flowise should confirm they are on version 3.1.0 or later to remediate CVE-2026-40933 regardless of network exposure [5].

Short-Term Mitigations

Beyond closing individual exposures, security and platform teams should audit workflow and agent-builder credential handling, since the n8n token findings demonstrate that leaked API tokens and exposed management consoles compound each other. Rotating any API tokens or credentials that may have been embedded in workflow exports, configuration files, or code committed to public or semi-public repositories should be treated as a priority parallel workstream to network remediation. Teams should also apply TLS termination and authentication at the reverse proxy layer for any self-hosted AI service, run these services under least-privilege, non-root accounts, and enable structured logging so that anomalous access attempts against newly secured endpoints can be detected. Vendor hardening guidance — such as Open WebUI's own documentation on securing public-facing deployments — should be treated as a mandatory deployment checklist rather than optional reading [4][7].

Strategic Considerations

Organizations should treat any AI infrastructure component that stores credentials, connects to internal systems, or holds proprietary data (embeddings, documents, conversation logs) with the same operational discipline applied to a password manager or secrets vault: default-deny network exposure, mandatory authentication, and regular access review. Because self-hosted AI tools are often deployed by individual developers or teams outside formal change-management review — a pattern consistent with, though not directly measured by, the scanning research cited here — organizations should extend existing cloud and SaaS asset-inventory programs to explicitly capture self-hosted AI infrastructure, and should consider periodic external attack-surface scanning — using the fingerprinting approaches demonstrated by this research and by tools such as AIMap — as a recurring control rather than a one-time exercise [1][6].

CSA Resource Alignment

This research note extends findings from two recent CSA AI Safety Initiative advisories that documented closely related exposure patterns in adjacent layers of the AI infrastructure stack. *LiteLLM AI Gateway: KEV-Listed Attack Chain Enables Full Takeover* analyzed how an internet-exposed, unauthenticated LiteLLM gateway became the entry point for a CISA KEV-listed, actively exploited remote-code-execution chain, and its recommendations on binding AI infrastructure to trusted networks, enforcing authenticated reverse proxies, and applying least-privilege service accounts apply directly to the model servers and agent platforms documented here [2]. *MCP Security Crisis: Systemic Design Flaws in AI Agent Infrastructure* documented a parallel exposure problem specific to Model Context Protocol servers, finding that the MCP authorization specification treats authentication as optional and that internet scanning has repeatedly identified thousands of publicly accessible MCP instances exposing full tool listings without credential validation — the same structural pattern of convenience-oriented insecure defaults documented in this note across Ollama, vLLM, Open WebUI, and agent-builder deployments [3]. Collectively, these findings map most directly to the AI Application Security and Identity and Access Management domains of CSA's AI Controls Matrix (AICM v1.1), which establishes control expectations for authentication, network exposure, and least-privilege configuration of AI systems and should serve as the baseline reference framework for organizations remediating the exposure documented in this note [8].

References

[1] Paganini, Pierluigi. "The AI supply chain has a security problem, and much of it is sitting on the open internet." Security Affairs, September 11, 2026.

[2] Cloud Security Alliance. "LiteLLM AI Gateway: KEV-Listed Attack Chain Enables Full Takeover." CSA AI Safety Initiative, June 17, 2026.

[3] Cloud Security Alliance. "MCP Security Crisis: Systemic Design Flaws in AI Agent Infrastructure." CSA AI Safety Initiative, May 4, 2026.

[4] Open WebUI. "Hardening Open WebUI." Open WebUI Documentation, 2026.

[5] SentinelOne. "CVE-2026-40933: Flowiseai Flowise RCE Vulnerability." SentinelOne Vulnerability Database, 2026.

[6] Zorz, Mirko. "AIMap: Open-source tool finds and tests exposed AI endpoints." Help Net Security, May 6, 2026.

[7] Twingate. "Run Ollama and Open WebUI Securely, No Open Ports." Twingate Blog, 2026.

[8] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." CSA AI Safety Initiative, 2026.

